Supervised learning

Lecture – 2 (GNR 652)

Biplab Banerjee

Supervised learning

Given:

- a set of input features $X_1, \ldots, X_n$
- A target feature Y
- a set of training examples where the values for the input features and the target features are given for each example
- a new example, where only the values for the input features are given

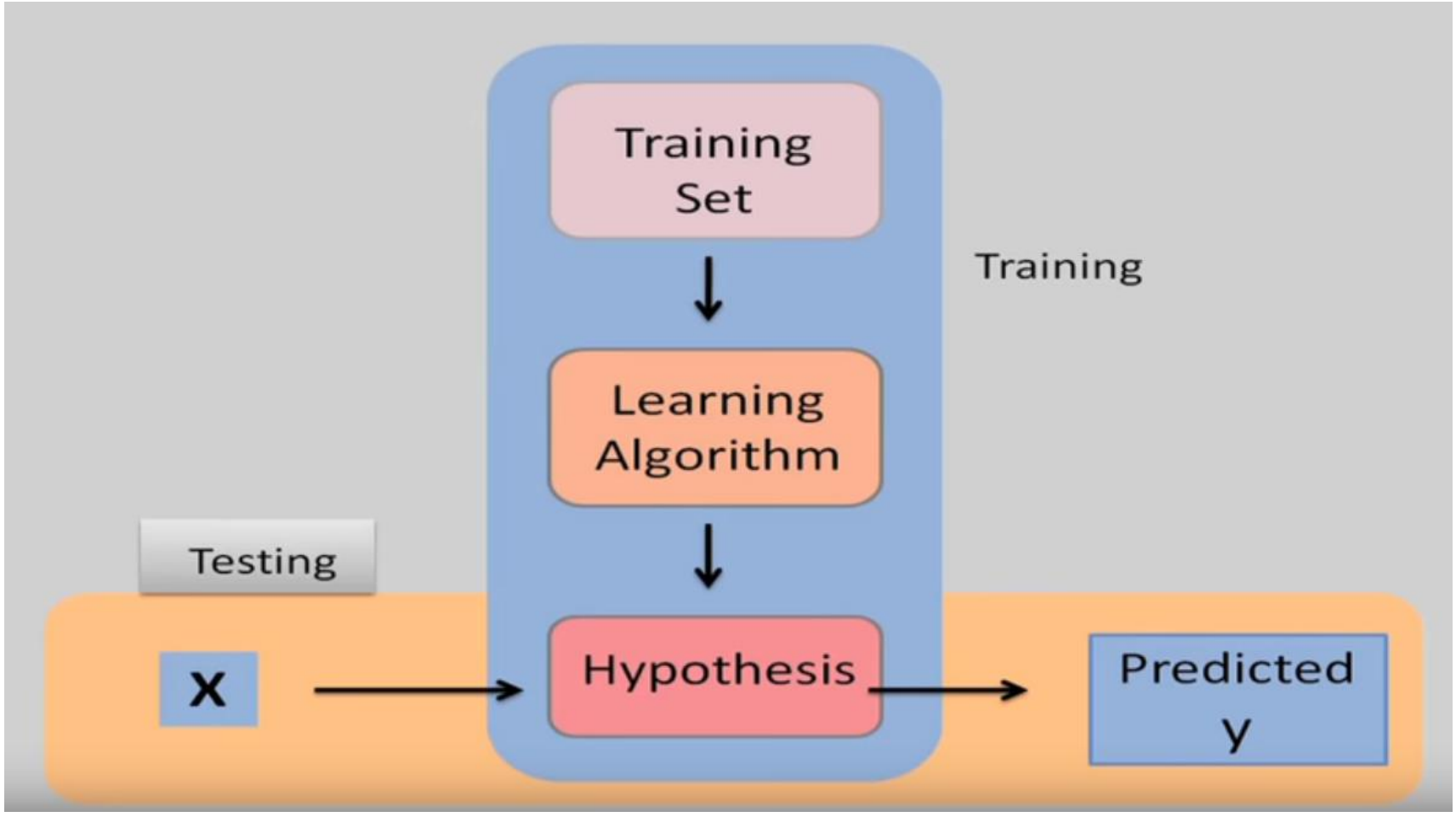
Predict the values for the target features for the new example.

- classification when Y is discrete
- regression when **Y** is continuous

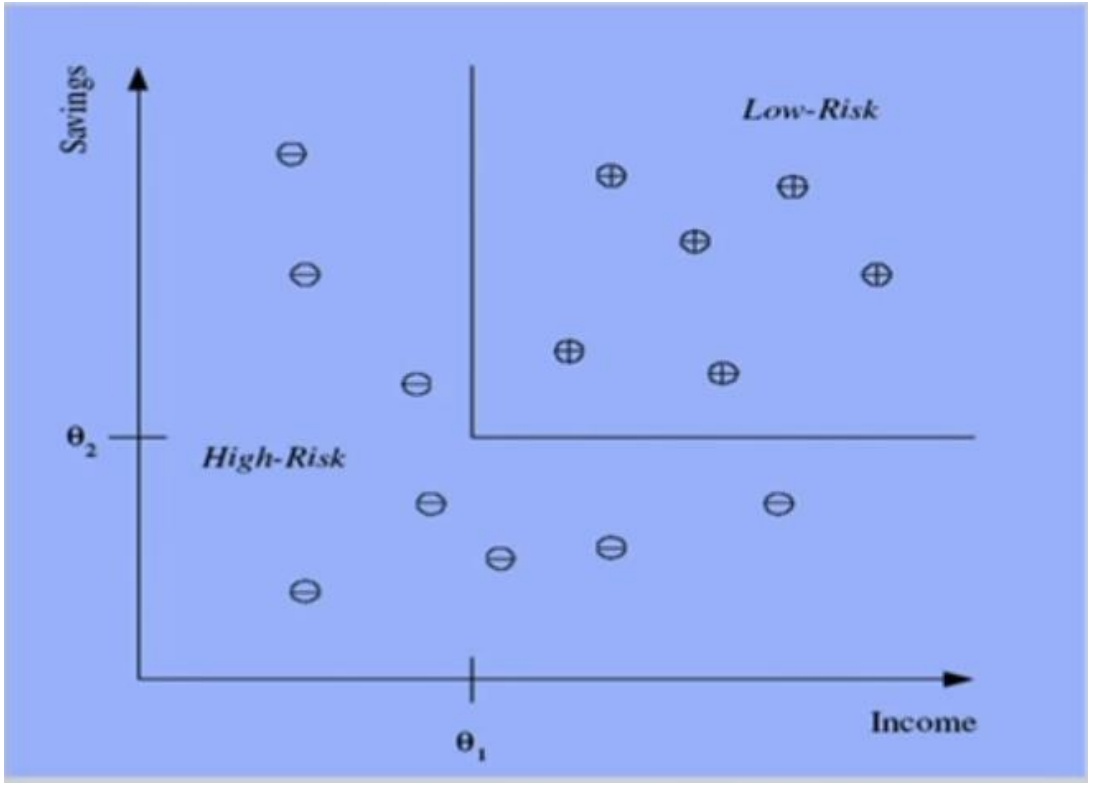
Supervised learning

we care about performance on the *distribution*, not the *training data*

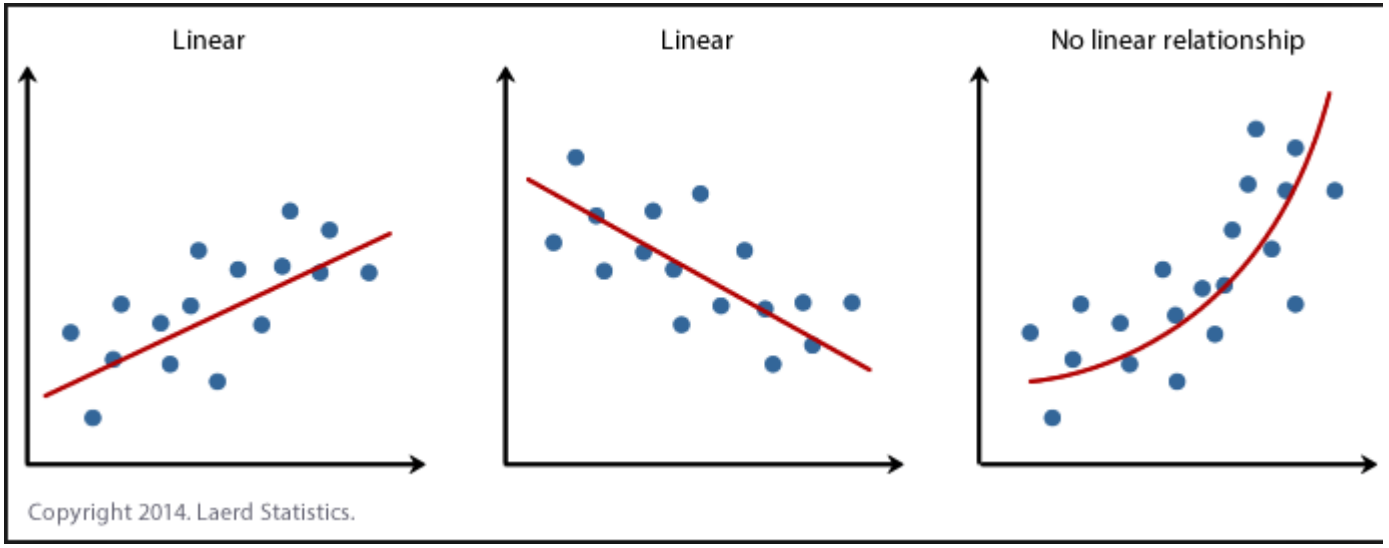
- Task *T:*
  - input: a set of *instances* $d_1,\ldots,d_n$
  - output: a set of *predictions* $\hat{y}_1,\ldots,\hat{y}_n$
- Performance metric *P:*
  - Prob (wrong prediction) on examples from *D*
- Experience *E:*
  - a set of *labeled examples (x,y)* where *y* is the true label for *x*
  - ideally, examples should be *sampled* from some fixed distribution *D*


Training Set
Training
Learning Algorithm
Testing
x
Hypothesis
Predicted y

Example - classification



Example - regression



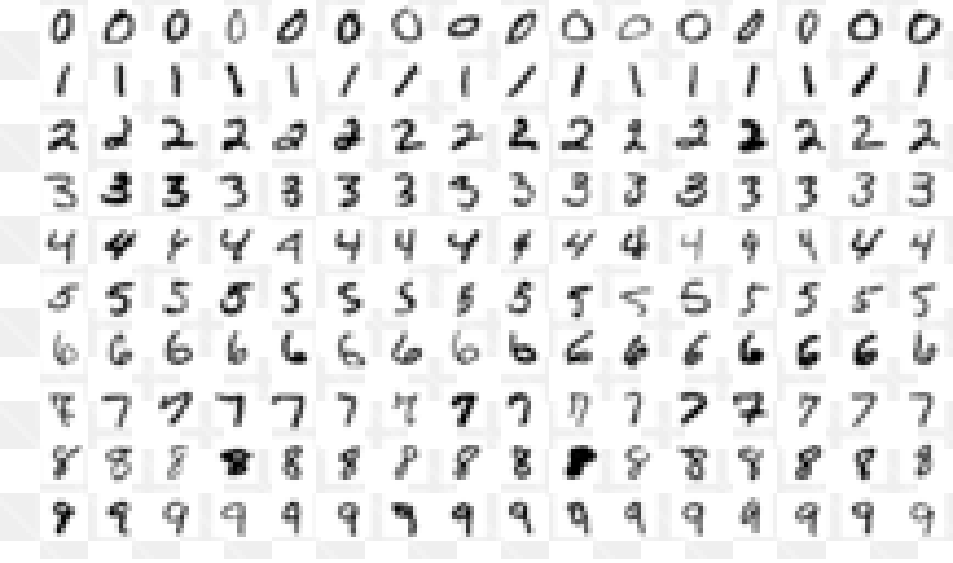
Let’s look into features

- Categorical
- Ordinal
- Real-valued
- Integer-valued

Some examples

Task	Instance	Labels
medical diagnosis	**patient record:** blood pressure diastolic, blood pressure systolic, age, sex (0 or 1), BMI, cholesterol	{-1,+1} = low, high risk of heart disease
finding company names in text	**a word in context:** capitalized (0,1), word-after-this-equals-Inc, bigram-before-this-equals-acquired-by, …	{first,later,outside} = first word in name, second or later word in name, not in a name
brain-human-interface	**brain state**: neural activity over the last 100ms of 96 neurons	{n,s,e,w,ne,se,nw,sw} = direction you intend to move the cursor
image recognition	**image:** 1920*1080 pixels, each with a code for color	{0,1} = no house, house

MNIST



Vector space at a glance

- An ordered *n*-tuple :

  a sequence of *n* real numbers $(x_1, x_2, \cdots, x_n)$

- R^n-space :

  the set of all ordered *n*-tuples

$n = 1$ R^1-space = set of all real numbers
(R^1-space can be represented geometrically by the *x*-axis)

$n = 2$ R^2-space = set of all ordered pair of real numbers (x_1, x_2)
(R^2-space can be represented geometrically by the *xy*-plane)

$n = 3$ R^3-space = set of all ordered triple of real numbers (x_1, x_2, x_3)
(R^3-space can be represented geometrically by the *xyz*-space)

$n = 4$ R^4-space = set of all ordered quadruple of real numbers (x_1, x_2, x_3, x_4)

u,v – vectors, c,d - scalars

(1) $\mathbf{u}+\mathbf{v}$ is a vector in R^n (closure under vector addition)

(2) $\mathbf{u}+\mathbf{v} = \mathbf{v}+\mathbf{u}$ (commutative property of vector addition)

(3) $(\mathbf{u}+\mathbf{v})+\mathbf{w} = \mathbf{u}+(\mathbf{v}+\mathbf{w})$ (associative property of vector addition)

(4) $\mathbf{u}+\mathbf{0} = \mathbf{u}$ (additive identity property)

(5) $\mathbf{u}+(-\mathbf{u}) = \mathbf{0}$ (additive inverse property)

(6) $c\mathbf{u}$ is a vector in R^n (closure under scalar multiplication)

(7) $c(\mathbf{u}+\mathbf{v}) = c\mathbf{u}+c\mathbf{v}$ (distributive property of scalar multiplication over vector addition)

(8) $(c+d)\mathbf{u} = c\mathbf{u}+d\mathbf{u}$ (distributive property of scalar multiplication over real-number addition)

(9) $c(d\mathbf{u}) = (cd)\mathbf{u}$ (associative property of multiplication)

(10) $1(\mathbf{u}) = \mathbf{u}$ (multiplicative identity property)

Subspace

$(V, +, \cdot)$: a vector space

$\left.\begin{array}{l} W \neq \Phi \\ W \subseteq V \end{array}\right\}$: a nonempty subset

$(W, +, \cdot)$: The nonempty subset W is called a subspace **if** W **is a vector space** under the operations of addition and scalar multiplication defined in V

Ideas of

- Linear independence
- Span
- Basis

We are interested about the Euclidean space

Metric space

A **metric** on a set S is a function $d : S \times S \to \mathbb{R}$ that satisfies

(i) $d(x, y) \geq 0$, with equality if and only if $x = y$

(ii) $d(x, y) = d(y, x)$

(iii) $d(x, z) \leq d(x, y) + d(y, z)$ (the so-called **triangle inequality**)

Normed space

A **norm** on a real vector space V is a function $\|\cdot\| : V \to \mathbb{R}$ that satisfies

(i) $\|\mathbf{x}\| \geq 0$, with equality if and only if $\mathbf{x} = \mathbf{0}$

(ii) $\|\alpha\mathbf{x}\| = |\alpha|\|\mathbf{x}\|$

(iii) $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$ (the **triangle inequality** again)

Any normed space is also a metric space!

Inner product space

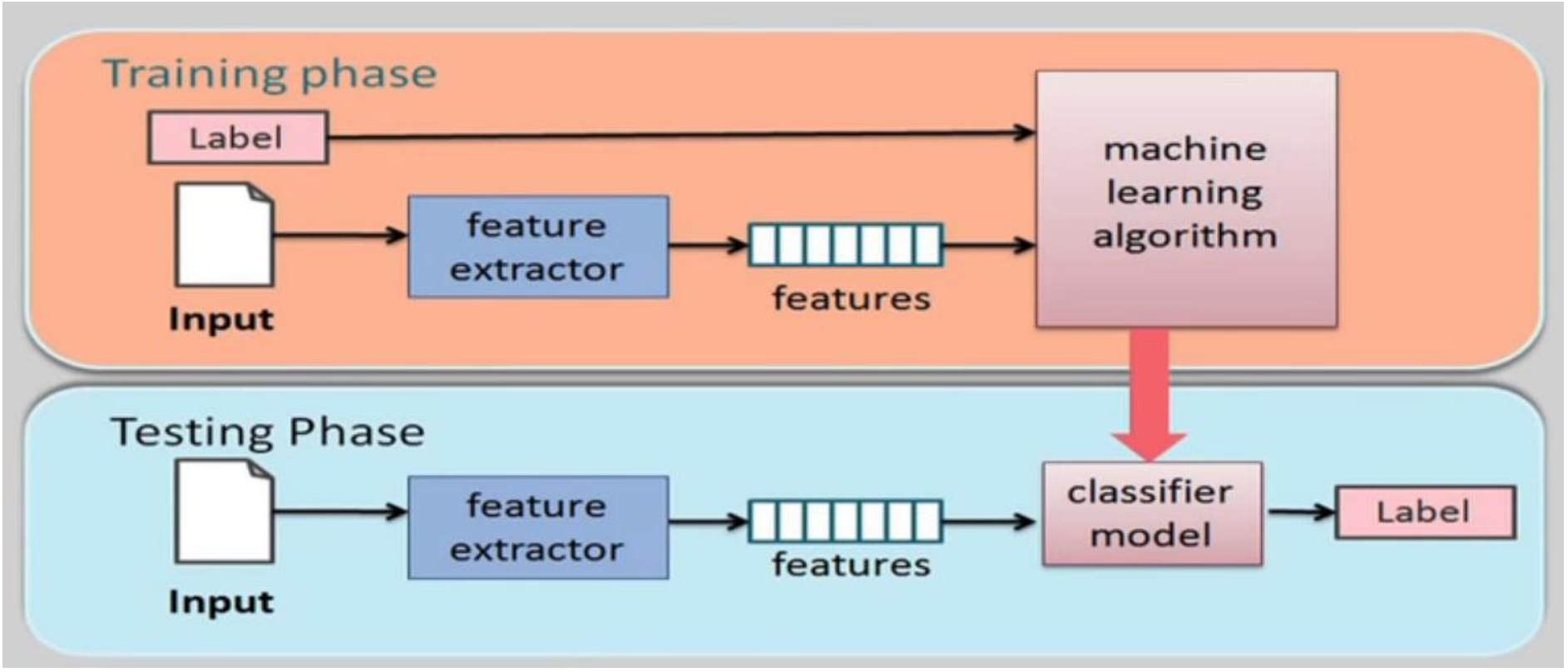
An **inner product** on a real vector space V is a function $\langle \cdot, \cdot \rangle : V \times V \to \mathbb{R}$ satisfying

(i) $\langle \mathbf{x}, \mathbf{x} \rangle \geq 0$, with equality if and only if $\mathbf{x} = \mathbf{0}$

(ii) Linearity in the first slot: $\langle \mathbf{x} + \mathbf{y}, \mathbf{z} \rangle = \langle \mathbf{x}, \mathbf{z} \rangle + \langle \mathbf{y}, \mathbf{z} \rangle$ and $\langle \alpha \mathbf{x}, \mathbf{y} \rangle = \alpha \langle \mathbf{x}, \mathbf{y} \rangle$

(iii) $\langle \mathbf{x}, \mathbf{y} \rangle = \langle \mathbf{y}, \mathbf{x} \rangle$

Supervised learning



In a nutshell

1. Choosing the training experience
   - Examples of best moves, games outcome …
2. Choosing the target function
   - board-move, board-value, …
3. Choosing a representation for the target function
   - linear function with weights (hypothesis space)
4. Choosing a learning algorithm for approximating the target function
   - A method for parameter estimation

Idea of concept

Subset of examples from the training data which uniquely represent a given concept/class.

- Examples: "Days at which my friend Aldo enjoys his favorite water sport"
- Result: classifier for days = description of Aldo's behavior

Sky	Temp	Humid	Wind	Water	Forecst	EnjoySpt
Sunny	Warm	Normal	Strong	Warm	Same	Yes
Sunny	Warm	High	Strong	Warm	Same	Yes
Rainy	Cold	High	Strong	Warm	Change	No
Sunny	Warm	High	Strong	Cool	Change	Yes

What is the general concept?

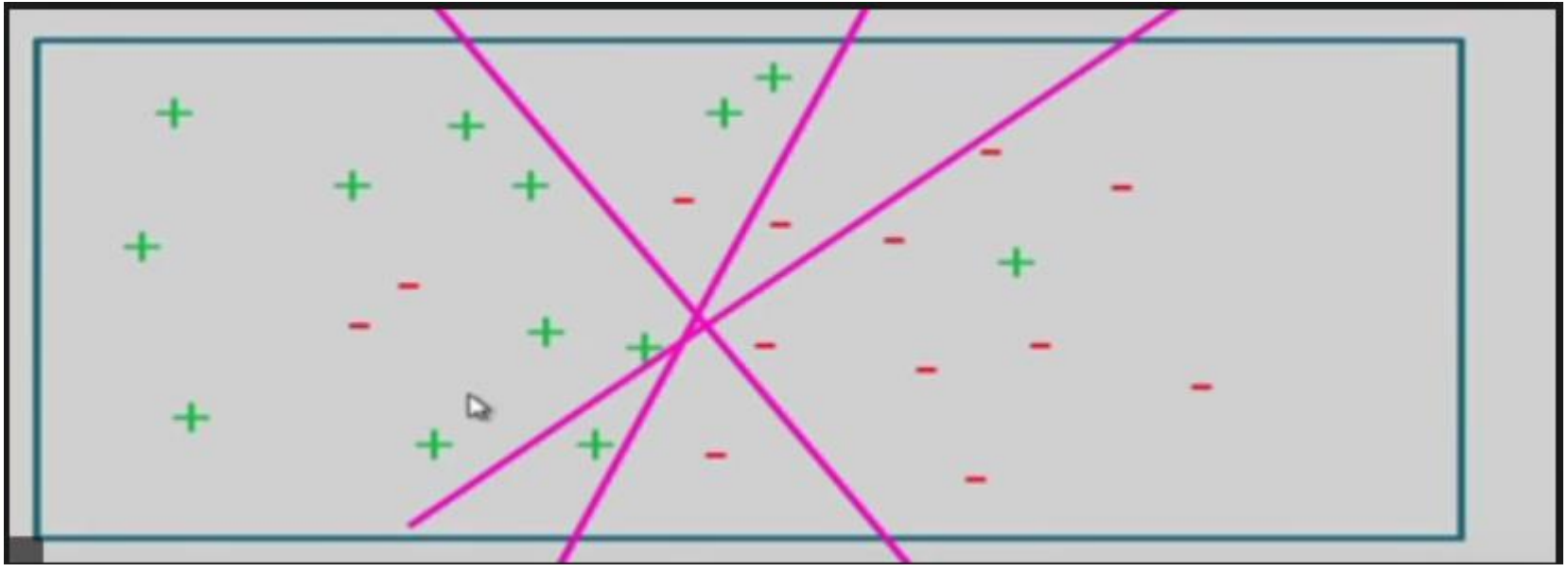
Hypothesis

- The function mapping which
  - Maps (hopefully) correctly the inputs to the outputs of the training set
  - Generalizes well to previously unseen examples
  - Approximates the true target function

Representation of the hypothesis language

- Decision tree
- Linear regressor
- Multi-level perceptron
- Margin based techniques
- ...

Many hypothesis - Hypothesis space



The task of concept learning

- Given:
  - Instances X: Possible days, each described by the attributes

    Sky, AirTemp, Humidity, Wind, Water, Forecast

  - Target function c: $EnjoySport : X \rightarrow \{0, 1\}$
  - Hypotheses H: Conjunctions of literals. E.g.

    $$\langle ?, Cold, High, ?, ?, ? \rangle.$$

  - Training examples D: Positive and negative examples of the target function

    $$\langle x_1, c(x_1) \rangle, \ldots \langle x_m, c(x_m) \rangle$$

- Determine: A hypothesis h in H with $h(x) = c(x)$ for all x in D.

Idea of version space

Definition A hypothesis h is consistent with a set of training examples D of target concept c if and only if $h(x) = c(x)$ for each training example $\langle x, c(x)\rangle$ in D.

$$Consistent(h, D) \equiv (\forall \langle x, c(x)\rangle \in D)\ h(x) = c(x)$$

Definition The version space, $VS_{H,D}$, with respect to hypothesis space H and training examples D, is the subset of hypotheses from H consistent with all training examples in D.

$$VS_{H,D} \equiv \{h \in H | Consistent(h, D)\}$$

What is a classifier then?

- Hypothesis h: Function that approximates f.
- Hypothesis Space $\mathcal{H}$: Set of functions we allow for approximating f.
- The set of hypotheses that can be produced, can be restricted further by specifying a language bias.
- Input: Training set $\mathcal{S} \subseteq X$
- Output: A hypothesis $h \in \mathcal{H}$

Inductive learning

The inductive learning hypothesis: Any hypothesis found to approximate the target function well over a sufficiently large set of training examples will also approximate the target function well over other unobserved examples.

Occum razor

Idea of inductive bias

- A classical example of Inductive Bias

- the simplest consistent hypothesis about the target function is actually the best

Hoeffding inequality

Relation between the error/ risk in the training and test samples.
Both the errrors converge with increasing ‘m’

$$P(|\varepsilon(h_i) - \hat{\varepsilon}(h_i)| > \gamma) \leq 2ke^{-2\gamma^2 m}$$